



Singularity

Galen Hunt
James Larus
David Tarditi

Microsoft Research

MSR TAB
June 21, 2005



General-purpose Computing Platform (PC)



- Software is difficult to install, maintain, and administer
- Applications interact in complex, unpredictable ways
- Almost no users understand computers or software and so react naively to unexpected behavior
- System administration is costly and unavailable to most



Talk Summary

- Advances in languages, compilers, and tools open the possibility of improving software
- Singularity uses these advances as a basis to build more reliable systems and applications
- Systems built on Singularity expand software delivery opportunities



Advances in Language & Compiler Technology

- Software advances enable better system architectures
 - more robust and verifiable than existing (40 year old) model

	Advance	
	Language safety	Program and system verification
Technology	expressive safe languages and type systems	explicit specifications and system descriptions
	optimizing compilers	defect detection/testing tools
	end-to-end safety checking	static program analysis
Microsoft	increasingly precise analysis systems with finer-grain isolation and fault-containment boundaries	improved static and dynamic error detection strong system guarantees



Singularity Project

- Develop technology and infrastructure to build more dependable software
 - language support to increase software quality
 - tools to ensure correct software behavior
 - OS architecture to enhance security, reliability, and effectiveness of tools
 - superior reliability, sufficient performance
- Large research project
 - current: ~dozen researchers & RSDs
 - Redmond, SVC, Cambridge
- System running on hardware and Virtual PC
- Not Windows successor!



Key Aspects of Singularity

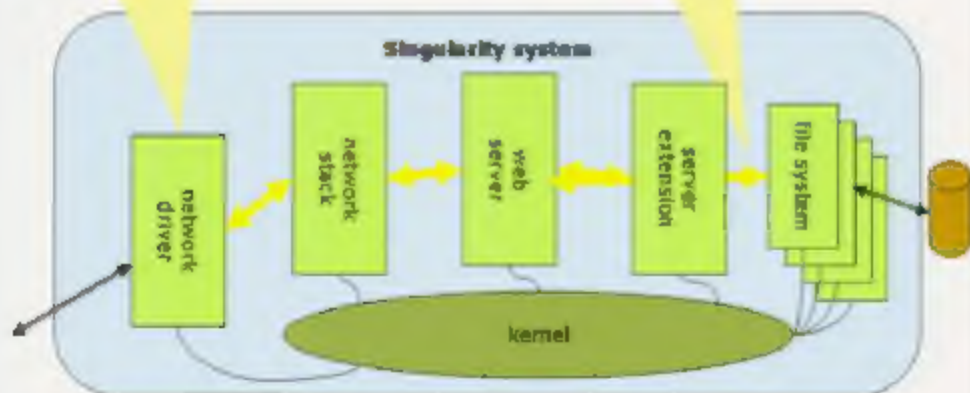
- Software-isolated processes (SIPs)
 - inexpensive isolation — memory, communications, failure — boundaries
 - OS manages and reclaims resources
 - “closed world” for program analysis
 - single isolation and extension model for all parts of application and system
- Merge OS and language runtime (VM/CLR)
 - prohibit unsafe code
 - remove duplicative APIs and security abstractions
 - fast, lightweight managed code run-time system
 - typed assembly language (TAL) reduces trusted computing base
- Language extensions to improve reliability
 - Spec#/Sing# specifications and verification
 - channel contracts
 - explicit and verified resource usage and reclamation



Singularity Architecture

SIP (software isolated process)

channel





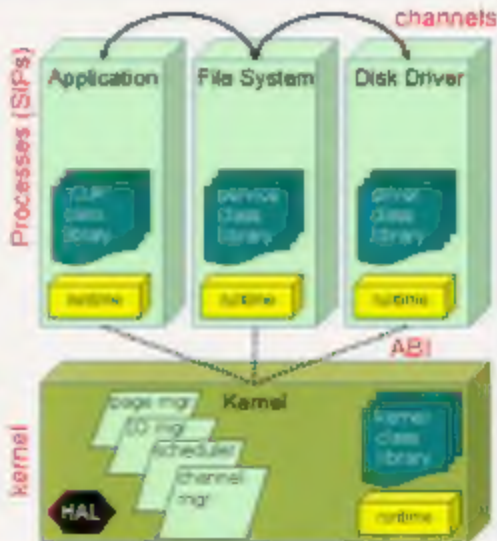
Presentation

- Introduction (Jim)
- **Singularity OS** (Galen)
- Language, compiler & runtime (David)
- Opportunities (Jim)



Detailed Architecture

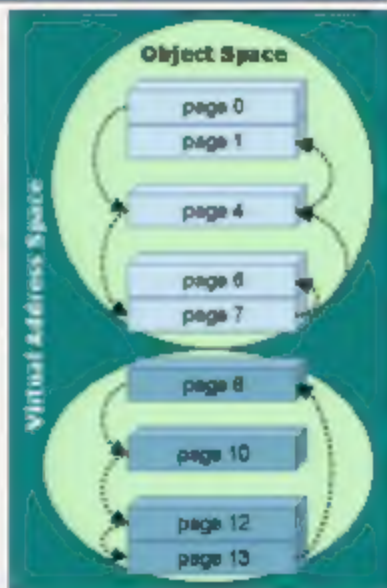
- Microkernel
 - apps, extensions, services, and drivers are all processes
 - HAL w/ PIC, RTC, timer, and console output
- Closed processes (SIPs)
 - no shared memory
 - no dynamic code loading
 - no dynamic code generation
- IPC via channels w/ contracts
- Abstract instruction set
 - type safe, memory safe MSIL
 - all third-party code is safe
 - layer of indirection
- Well-defined, strongly-versioned application binary interface (ABI)





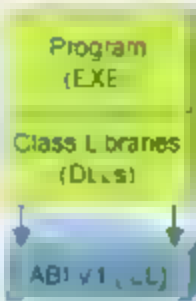
Software Isolated Processes (SIPs)

- Each SIP
 - contains verified user code
 - type safe
 - no privileged instructions
 - has exclusive ownership of pages
 - contains pointers to its pages only
 - self-contained for GC
 - has own runtime
- SIPs can
 - run in kernel address space
 - run in ring0
 - kernel hw protection domain

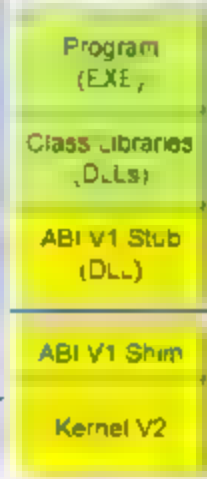


- Kernel access Bin. ready to execute and
 finished and
 running
 ready to accept call
- API as defined in the
 as a set of
 as a set of
- a single application to a file
 the file
- v1 API as a set of

Developer or



User PC



On X86-64

Kernel

- happens

- asynchronous bit transfer

- error, flow control

- slowly, like the old serial interface

- packet multiplex

- it's a good idea

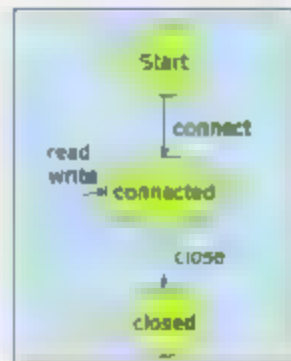
- Message may contain data and control

- Messages are handled in a way

- send removes message from sender

- receive inserts message to receiver

- at most, allowing connection at any time



Process

Channel

Process

Communications Heap

Process

Channel

Process

Communications Heap

Process

Channel

Process

Communications Heap

Cost (CPU Cycles)

	Singularity	FreeBSD	Linux	Windows
Alloc	1	1	1	44
Free	1	1	1	10
Exec	1	1	1	410
Wait	1	1	1	10
Read	1	1	1	10
Write	1	1	1	10
Copy	1	1	1	10
Process	1	1	1	10
Network	1	1	1	10
Storage	1	1	1	10
Security	1	1	1	10
Logging	1	1	1	10
Configuration	1	1	1	10
Monitoring	1	1	1	10
Reporting	1	1	1	10
Administration	1	1	1	10
Documentation	1	1	1	10
Support	1	1	1	10
Training	1	1	1	10
Research	1	1	1	10
Development	1	1	1	10
Testing	1	1	1	10
Deployment	1	1	1	10
Operation	1	1	1	10
Maintenance	1	1	1	10
Upgrade	1	1	1	10
Backup	1	1	1	10
Restore	1	1	1	10
Disaster Recovery	1	1	1	10
Compliance	1	1	1	10
Audit	1	1	1	10
Incident Response	1	1	1	10
Forensics	1	1	1	10
Legal	1	1	1	10
Public Relations	1	1	1	10
Marketing	1	1	1	10
Sales	1	1	1	10
Customer Support	1	1	1	10
Partners	1	1	1	10
Investors	1	1	1	10
Board	1	1	1	10
Shareholders	1	1	1	10
Employees	1	1	1	10
Contractors	1	1	1	10
Suppliers	1	1	1	10
Vendors	1	1	1	10
Regulators	1	1	1	10
Law Enforcement	1	1	1	10
Media	1	1	1	10
Academics	1	1	1	10
Industry	1	1	1	10
Government	1	1	1	10
Military	1	1	1	10
Intelligence	1	1	1	10
Healthcare	1	1	1	10
Education	1	1	1	10
Religion	1	1	1	10
Arts	1	1	1	10
Science	1	1	1	10
Technology	1	1	1	10
Environment	1	1	1	10
Energy	1	1	1	10
Transportation	1	1	1	10
Communication	1	1	1	10
Finance	1	1	1	10
Insurance	1	1	1	10
Real Estate	1	1	1	10
Construction	1	1	1	10
Manufacturing	1	1	1	10
Retail	1	1	1	10
Food	1	1	1	10
Health	1	1	1	10
Pharmaceuticals	1	1	1	10
Automotive	1	1	1	10
Aerospace	1	1	1	10
Defense	1	1	1	10
Space	1	1	1	10
Outer Space	1	1	1	10
Universe	1	1	1	10
Everything	1	1	1	10

- Buffer overflow and other

flaws

highlight, delete, copy, paste

- Singularity kernel

processes and memory mapped

- domains and security

- High level data structure for user and kernel

non-blocking

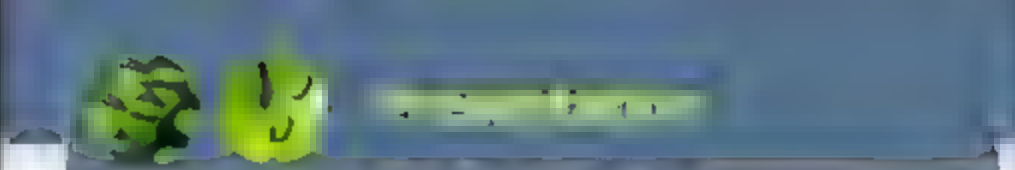
additional features and development

- New and old, long term

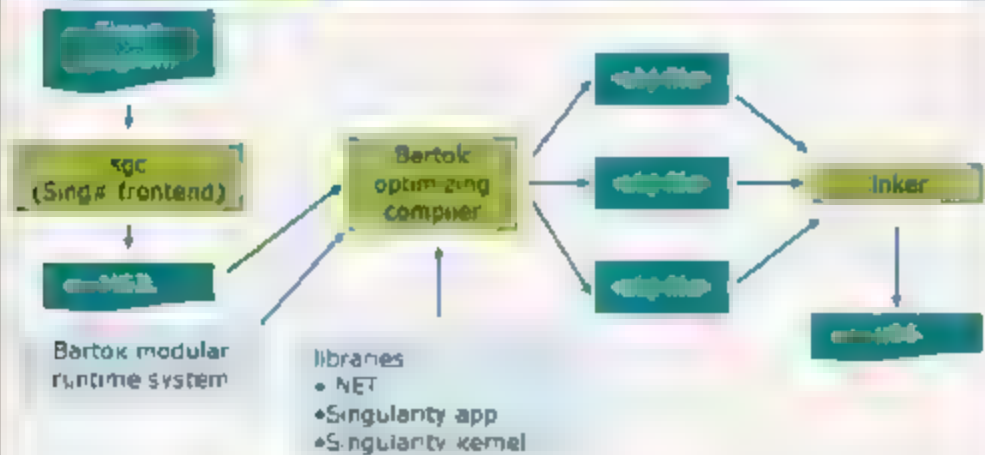
Microkernel and other

names and architecture manager

Buffer and Buffer management

- 
- Introduction
 - Singularity OS
 - **Language, compiler & runtime**
 - Opportunities

- Systems programming in safe garbage collected language
- Built on Bartok compiler & run time system and Specl# language
- Language extensions
 - for systems programming
 - for distributed systems programming
- Highly optimizing ahead of time compiler
- Small, customizable run time system
 - support for multi-processor architectures



Bartok targets Singularity and Windows



- System integrity depends on code safety
 — Time and memory safety, interfaces, I/O safety
- Deterministic execution rate



machine code
verification

byte code
verification

Singularity
TCB
proof
verification

- Currently, Singularity compiler ensures a B
- Working on sing. coded assembly language A, I, E, C, A, T, B

✓ reported

- `yaml` class adding
- `data` is very limited support
- `data` is very limited support
- `data` is very limited support
- `data` is very limited support

- Not based on references
- Channel objects
 - specify how message passing and data flow is achieved
 - allow implementation of a new channel
- Support for stateful and stateless channels
- Can be mapped to hardware or software
- With a new statement for specifying event pattern matching
- New data structures for managing resources
- Overlays for type checking and data flow analysis



ANSWER

- Problem efficient communications - through data sharing
- Answer java main method
data transferred via send to receiver
sender don't retain message once recd pr
permits efficient data passing
- Info security via stream
- java.io package contains all classes
PrintStream
PrintWriter
PrintStream
PrintWriter
- All streams are restricted to either
a) created inside of program
and don't interact with hardware
or b) be used after program ends
escape
- All streams are primarily responsible for packaging between processes

namespace sample

public ref struct

public int NumberOfItems

public int NumberOf...

Base
contract

public contract interface IContract

out message int

in message int

out message string

Message sent
by contract
exporter

override state int read, int write

state Ready state, etc. to ... etc. to ... Ready state

Message received

Message
sent

Next state

- Contract declares

message types and parameter types defined in
state machine according to its message sequences

- Provides

change state, stop, start, suspend, resume,
home, shutdown, method
lookup and state method

- Efficient

pre-allocated buffers
state machine send/receive messages

sample sample

rap struct

public int Name

public int Name

Base
contract

public contract state state

out message

in message Get

out message Set

Message sent
by contract
exporter

override state

state ready state

Message received

Message
sent

Next state



Computer Architecture

- Bartok supports whole program and separate compilation

separate compilation allows components to be compiled in parallel. May be done on a system level or an application level.

- Assists specialization of runtime systems

allows a user to generate a specialized runtime system for a particular application. The user can generate a specialized runtime system for a particular application.


- Extensive optimization

• The compiler can generate code that is more efficient than the original code. The compiler can generate code that is more efficient than the original code.

Journal of Management Inquiry

- Each process's runtime environment is autonomous

brancos, e a maioria dos negros, em 1998, em 2000 e em 2002.

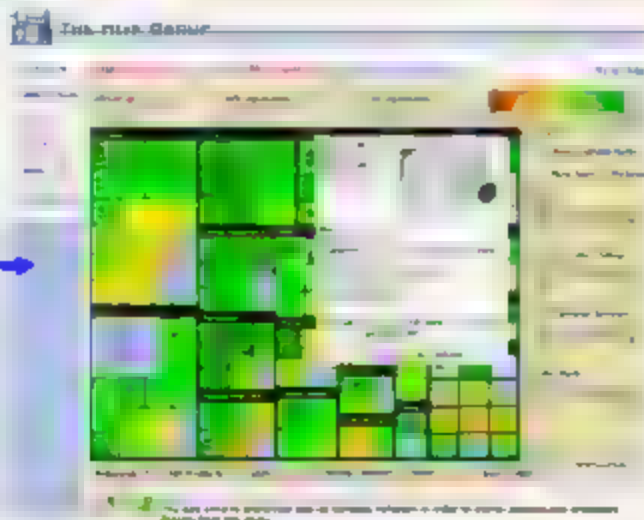
- 
- Introduction
 - Singularity OS
 - Language, compiler & runtime
 - Opportunities



PROJETO

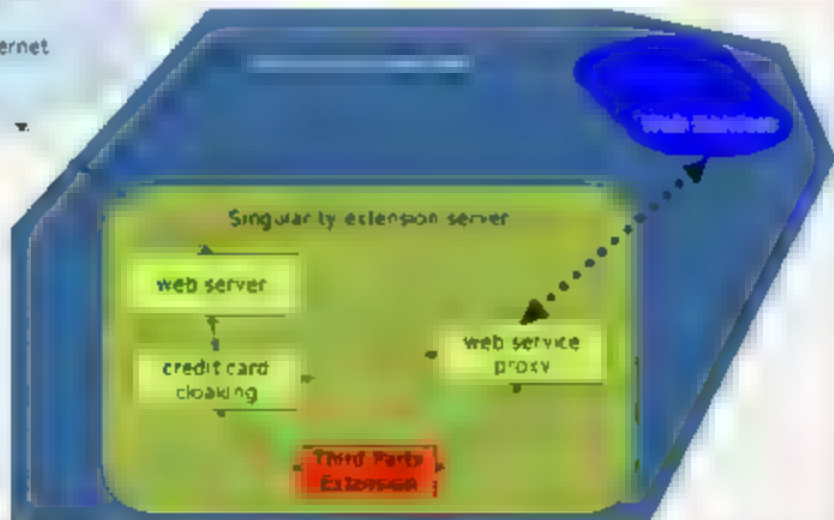
- Descreva o ambiente onde o projeto será realizado
- Numa palavra, descreva o projeto
- Descreva o resultado esperado
- Qual a importância do projeto para a empresa?
 -
 -
 -
 -
 -

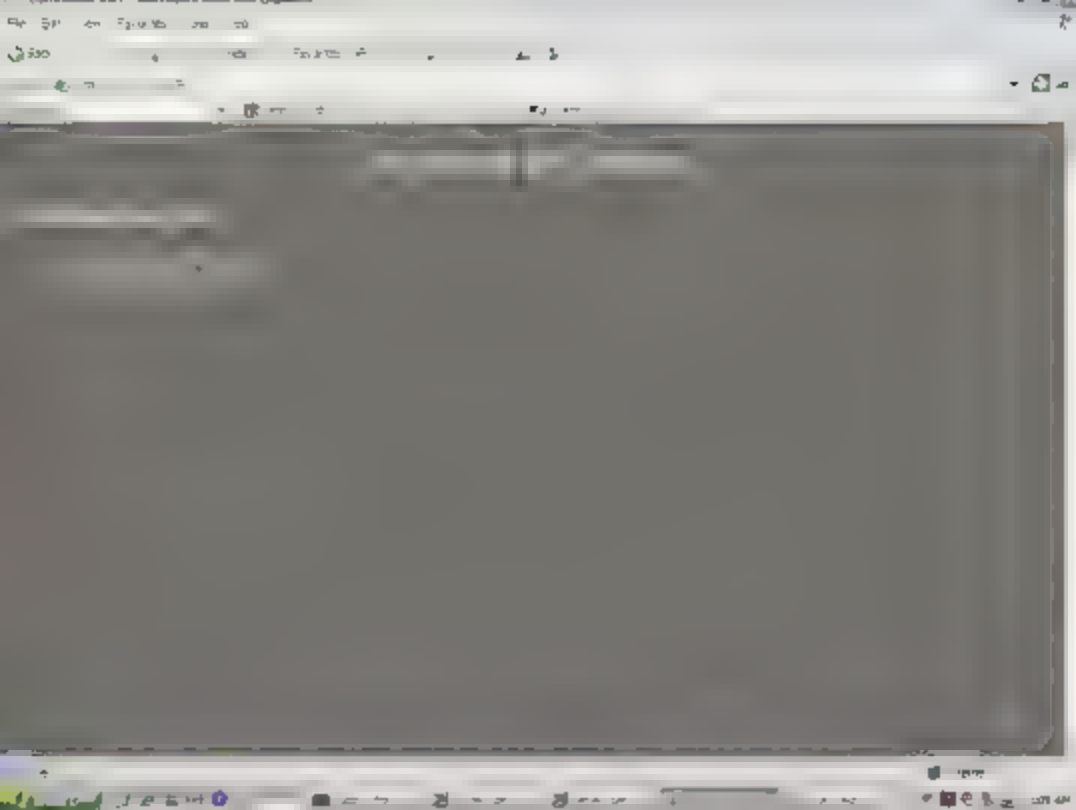
Amazon.com Dg



Amazon.com customer





Internet





In Search of Caffeine



Looks like you have quite a few choices.

-  [Starbucks](#)
-  [Pumpkin's Best Coffee](#)
-  [Blue](#)
-  [Coffee with a twist](#)
- [Take me to Restaurant nearby](#)



In Search of Caffeine

Looks like you have quite a few choices

-  Tulz
-  Tulz's with wireless
- [More all choices](#)
- [Take me to Redmond instead](#)





Conclusion

- Singularity enables new systems and applications
 - started with: "what would a system that favored reliability over performance look like?"
 - building platform, tools, and technology to achieve this goal of improving software reliability
 - resulting system enables new software delivery vehicles and businesses
- Open research questions
 - verifiable programming abstractions
 - increased typed assembly language expressiveness
 - general error-handling and recovery mechanisms
 - more specification and better static defect detection tools
 - simpler, robust security model
 - first class application abstraction



Questions?

- <http://singularity>